

## A FORTRAN—PZ PROGRAMOZÁSI NYELV

Írta: PUSKÁS ALBERT

A dolgozat a *FORTRAN* nyelv egy változatát, a *FORTRAN—PZ* nyelvet definiálja. A PZ jelet a „Programozási nyelv Zászlós-ábrákkal való definiálása” rövidítésére használjuk. Célunk a változat kidolgozásával az volt, hogy megadjunk egy olyan programozási nyelvet, mely véleményünk szerint elegendő arra, hogy a számítástechnikai alapképzésben résztvevők (számítógéppel csak közvetett kapcsolatban levők) számára egy programozási nyelv felépítése, szerkezete ismertté és könnyen elsajátíthatóvá váljon.

„A programozási nyelvek — hasonlóan minden más nyelvhez — az információátvitel célját szolgálják valamely adó és vevő között (természetesen kódolt formában). A programozási nyelvek esetében az adó a feladatot leíró ember, a vevő pedig a feldolgozást végző elektronikus számítógép. Ez a körülmény a programozási nyelvek számos sajátosságát meghatározza, és egyben korlátozza lehetőségeiket. Mivel adóként az ember szerepel, célszerű lenne az emberéhez minél közelebb álló nyelvet alkalmazni az információátvitelre. Ugyanakkor biztosítani kell, hogy a vevő (a számítógép) képes legyen dekódolni és értelmezni minden számára szóló kódolt információt. Ez a két követelmény némiképpen ellentmond egymásnak, minthogy a gép számára közvetlenül érthető nyelv (saját belső kódrendszere) igen távol áll az emberi nyelvektől” [1].

A számítógépek programozásának első időszakában a programok felírása a gép kódrendszerében történt. Ez a tény erősen akadályozta alkalmazhatóságukat és elterjedésüket, annak ellenére, hogy a gépi kódrendszerek is jelentős fejlődésen mentek keresztül. Igény merült fel tehát olyan közvetítő nyelvek iránt, melyek közelebb állnak a természetes nyelvekhez; valamint az iránt, hogy a számítógépeket alkalmassá lehessen tenni ilyen közvetítő nyelv-(ek)-en közölt információk megértésére.

Ezen cél elérése érdekében az lett volna várható, hogy az ember és a gép közötti kommunikálás eszköze egyetlen „világnyelv” lesz, de egyáltalán nem ez történt, hanem meglepően nagyszámú mesterséges gépi nyelv fejlődött ki, melyek között vannak világnyelvek és vannak szerény dialógusok [5].

## A FORTRAN—PZ nyelv megalkotásának indokai

A *FORTRAN* nyelv változatának kidolgozásához elsősorban oktatási tényezők vezettek le. Azon indokainkat, melyek szükségessé tették a *FORTRAN—PZ* nyelv megkonstruálását, létrehozását, elsősorban a megtanulhatóság, a megtaníthatóság fő problémája köré csoportosítjuk, figyelembe véve, hogy a 70-es években a számítástudomány már annyira része, illetve eszköze az alkalmazott tudományoknak, hogy a leendő felhasználók a közép- vagy alsóbb fokú iskolákban megtanulhatják a szükséges alapismereteket éppúgy, mint a matematikai alapokat, tehát az újabb generációknak egyáltalán nem lehet idegen egy egyszerű programozási nyelv notációja és megfogalmazás módja [5].

A számítástechnikai oktatás során a gépi programozás alapjául nyilván a gépi kódú programozás szolgál. A programozáshoz szükséges alapfogalmak, fogalmak, program készítési módok ismertetése itt válik lehetővé. Az egyszerű programozás azonban a gépi utasításrendszerrel szinte lehetetlen. A gépi kódú utasítások formája igen messze áll a hagyományos írásmódtól. A számítások során hagyományosan algebrai kifejezéseket használunk. A gépi kódú utasítások inkább a függvényírásmódot alkalmazzák (a műveleti kód a függvény-név, a címek az argumentumok). A hagyományosan írt számítási eljárás kifejezések (több műveleti jel és zárójel használatával) értékének kiszámításával hajtódik végre. Ezzel szemben a gépi kódú program egy-egy műveleti jelnek megfelelő utasításokra bomlik. Mindezek ellenére a gépi programozás oktatásában a gépi kódú programozás elengedhetetlen. Véleményünk szerint a hátrányok kiküszöbölésre a mnemonikus írásmód segítségével némileg lehetséges. Egyetemi és főiskolai előadások során sokkal hatékonyabbnak véljük a fiktív gépi utasításrendszerrel való programírást, mint valamely konkrét gépi utasításrendszerével. Bár több egyetem és főiskola az utóbbi utat választja, a tanárképző főiskolákon (elsősorban a szegedin) mi mégis a *Kalmár-féle* fiktív gép utasításrendszerének egy speciális változatát alkalmazzuk az 1963-ban tantervbe iktatott „A matematika modern alkalmazásai” c. tantárgy előadásai során.

Ma már nem állhat meg a gépi programozás oktatása (még az alapfokú képzésben sem) a gépi kódok szintjén, szükség van egy magasabb szintű programozási nyelv oktatására is.

A programozási nyelvek program írásmódja már lényegesen közelebb került a hagyományos írásmódhoz. A programozási nyelvekben felhasznált konstansok, operanduszok, változók, alaputasítások és vezérlő utasítások formája és struktúrája a szokásos matematikai formához és struktúrához már nagyon hasonló, de különösen strukturális felépítésében még lényegesen különböző is.

Az előbb említett programozási nyelvi elemeken kívül vannak a programozási nyelvekben olyan általános szabályok, utasítás-csoportok, technikai fogások, amelyek egyrészt a programozó munkáját könnyítik meg, másrészt a számítógépet látják el a végrehajtáshoz szükséges információkkal. Vannak továbbá az egyes nyelveknek (saját céljaira fenntartott) különleges jelentésű kulcsszavai, melyek általában angol szavak, ezek a nyelv alapjelei közé tartoznak és jelentésüket rendszerint magukban hordozzák. Ezen szabályok (és kulcsszavak is) a programozási nyelveket idegenszerűvé, bonyolulttá, nehezkessé teszik, miáltal formalizmusuk ismét távol kerül a matematikai jelölésmódtól.

Ezen okok miatt is szükségesnek tartjuk (a programozási nyelvek alapelemeinek szűkebb megtartása mellett) egy könnyebben megérthető, könnyebben megtanulható és könnyebben oktatható nyelv kidolgozását.

Másik, nem kevésbé fontos, indokunk a számítástechnikai program megvalósításának következménye.

A számítástechnikai program szerinti szakemberképzés intézményes oktatási részét az Művelődési Minisztérium vállalta magára. A képzés koncepciójának, célkitűzéseinek és programjának kidolgozására Számítástechnikai Bizottságok jöttek létre. Az M. M. Pedagógusképző Osztálya által létrehozott Albizottság dolgozta ki a pedagógusképző intézmények részére a *számítástechnikai képzés irányelveit*. Ezen irányelveket meghatározta az a tény, hogy a végzett matematika szakos hallgatók fogják oktatni az általános iskolákban azt a matematikát, mely hivatott arra is, hogy a tanulók a számítástechnika elemeivel megismerkedjenek. Így hallgatóinkat általános alapképzésben kell részesíteni, általános alapképzésen értve „olyan számítástechnikai ismeretanyag közlését, melyre támaszkodva a népgazdaság legkülönbözőbb területein

dolgozó szakemberek szakmai tevékenységüket a számítástechnika adta lehetőségek ismeretében tudják elvégezni” (MM Számítástechnikai Irányelvei, 1970). Az irányelveket két dolog determinálta: egyrészt az általános iskolai számítástechnikai képzés anyaga, másrészt a főiskola új tanterve adta lehetőségek [3].

Az említett irányelvek alapján az 1970/71. tanévtől bevezetett új főiskolai tantervben a matematika szakos hallgatók számára egy, a számítástechnikai alapképzést szolgáló új tárgy, a „*Numerikus és gépi módszerek*” c. tárgy került bevezetésre. E három féléves tárgy programja, többek között, előírja egy programozási nyelv ismertetését, szabad választást engedve az *ALGOL* és *FORTRAN* között. A szegedi Tanárképző Főiskolán két éven át a *FORTRAN*, az 1973/74. tanévben, mivel a módosított tanterv miatt a tárgy oktatása két évfolyamon is folyt, mindkét nyelv került előadásra. Az ezen anyagrésze szánt kevés óraszám (előadásra mintegy 18—20 óra, gyakorlatra pedig 8—10 óra áll rendelkezésre) indokolja, hogy nem kerülhet sor egy programozási nyelv teljes ismertetése. Az előadásokon, a gyakorlatokon és a vizsgákon szerzett többéves tapasztalat összegyűjtésével, felhasználásával hoztuk létre e szűkített változatot, mely véleményünk szerint a fent említett hallgatóság, de az alapképzésben bárhol részt vevők, számára elegendő egy programozási nyelv elemeinek és struktúrájának megismerésére.

Az új tanterv egy másik lehetőséget is ad a hallgatók számítástechnikai képzésére a tantervben szereplő kötelezően választható kollégiumok keretén belül. E két féléves tárgyak időben követik a már említett *Numerikus és gépi módszerek* három félévét. A kötelezően választható kollégiumok számítástechnikával foglalkozó része így lehetővé teszi, hogy egy programozási nyelv megadását metanyelvi módszerrel is tárgyalhassuk. Ilyen kollégiumok tananyagába is beépíthető e szűkített változat. Bár ennek oktatási sikerességére vonatkozó helyi tapasztalataink még nincsenek, hiszen ilyen kollégiumot még csak az 1974/75. tanévre hirdettük meg (az 1973/74. tanévvégi maximális érdeklődés már is sikernek tekinthető), azonban figyelembe véve a JATE matematikus képzése során szerzett tapasztalatokat, dolgoztuk ki a nyelv szintaxisát zászlós-ábrák segítségével. Erről az 1974-ben rendezett visegrádi számítástechnikai oktatással kapcsolatos konferencián részben beszámoltunk [4].

### A zászlós-ábrázolásról

KALMÁR LÁSZLÓ akadémikus „An intuitive representation of context — free languages” című cikkében [2] hangsúlyozza, hogy egy programozási nyelv szintaxisának megadására, az általa használt zászlós-ábrák előnyösen használhatók oktatói segédeszközként a programozási nyelvek tanításában. Ezzel egyetértve, a *FORTRAN*—*PZ* nyelv szintaxisát mi is zászlós-ábrákkal adjuk meg.

Ezért először összefoglaljuk a zászlós-ábrázoláshoz szükséges fogalmakat, és megadjuk azok értelmezését.

Mindenek előtt a kontextus-mentes nyelvek definíciójához jutunk el, majd leírjuk az ilyen nyelvek zászlós-ábrával való reprezentációját.

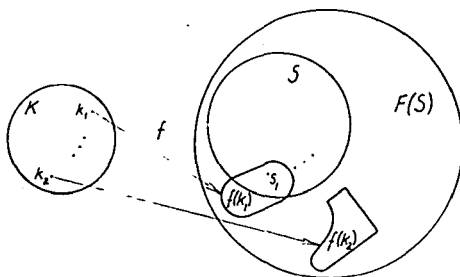
Először megadjuk a nyelvnek általános, általunk továbbiakban használt fogalmát:

Definíció: Egy  $L$  nyelv alatt egy  $L = \{S, K, f\}$  rendezett hármast értünk ahol,

1.  $S$  és  $K$  idegen, nem üres, véges halmazok,
2.  $f$  pedig  $K$  egy leképezése az  $S$  által generált  $F(S)$  szabad félcsoport összes részhalmazainak halmazába.

Megjegyzés: az  $F(S)$  szabad félcsoport művelete alatt a iuxtapozíciót (egymás után helyezést) értjük.

A definíciót az 1. ábra szemlélteti:



1. ábra 1.

Bevezetjük a következő elnevezéseket:

1. Az  $S$  halmazt *szókészletnek* (terminális szótár), elemeit pedig *szavaknak*,
2. a  $K$  halmazt *kategória-halmaznak*, elemeit *kategóriáknak* nevezzük.
3.  $F(S)$  a *szósorozatok* (szóláncok) halmaza.
4. Végül, ha valamely  $k \in K$  kategóriára  $f(k) \subseteq F(S)$  teljesül, akkor azt mondjuk, hogy  $f(k)$  a  $k$  *kategóriához tartozó szósorozat* (szósorozatok).

Ezek után a kontextus-mentes nyelv definiálásához szükséges néhány segédfogalmat vezetünk be.

Definíció: egy  $G$  kontextus-mentes grammatikán egy  $G = \{S, K, R\}$  *rendezett hármast* értünk, ahol

1.  $S$  és  $K$  idegen, nem üres, véges *halmazok*,
2. és  $R$  *részhalmlaza* a  $\{K \times F(S \cup K)\}$  halmaznak.

Az  $S$  és  $K$  halmazokat továbbra is *szókészletnek*, illetve *kategória-halmaznak*, az  $R$  halmazt pedig a *szabályok-halmazának* nevezzük. Az  $R$  halmaz  $r = (k, \varphi)$  elemei (ahol  $k \in K$  és  $\varphi \in F(S \cup K)$ ) a *szabályok*. Egy  $r = (k, \varphi)$  szabályt a továbbiakban  $k : \varphi$  módon jelölünk.

Most már egy  $k$  kategória fogalom *kifejtését* a következő módon értelmezhetjük:

- a) A  $k$  kategória *közvetlen* kifejtésének nevezzük az  $F(S \cup K)$  halmaz egy  $\varphi$  elemét, ha  $k : \varphi$  szabály.
- b) A  $k$  kategória *kifejtésének* nevezzük az  $F(S \cup K)$  halmaz egy  $\varphi$  elemét, akkor
  1. ha  $\varphi$  közvetlen kifejtése  $k$ -nak, vagy
  2. ha  $\varphi = \varphi_1 \varphi_2 \varphi_3$  alakú, továbbá ha  $\varphi' = \varphi_1 k' \varphi_3$  kifejtése  $k$ -nak és  $k' : \varphi$  szabály, akkor  $\varphi$  kifejtése  $k$ -nak.
- c) Végül a  $k$  kategória *terminális* kifejtésén a  $k$  kategória azon kifejtését értjük, melynek elemei csak az  $F(S)$  halmazból valók (azaz olyan sorozatok, melyeket csak szavakból képezünk).

A fenti segédfogalmak bevezetése után megadhatjuk a *kontextus-mentes nyelv definícióját*:

Akkor mondjuk, hogy egy  $L = \{S, K, f\}$  nyelvet a  $G = \{S, K, R\}$  kontextus-mentes grammatika generál, ha bármely  $k \in K$  kategóriára  $f(k)$  a  $k$  kategória összes terminális kifejtésének halmaza. Ezek után:

egy  $L = \{S, K, f\}$  nyelv akkor *kontextus-mentes*, ha valamely kontextus-mentes grammatika generálja.

Most rátérünk a kontextus-mentes nyelvek zászlós-ábrázolására. Először a zászlós-ábra értelmezését adjuk meg, majd leírjuk a szükséges jelölésrendszert, végül megmondjuk, hogy egy nyelv hogyan reprezentálható velük.

Definíció: Zászlós-ábrán egy  $A = \{S, K, G, f_1, f_2, g_1, g_2\}$  rendezett hetest értünk, ahol

1.  $S$  és  $K$  idegen, nem üres, véges halmazok,
2.  $G$  egy véges, irányított gráf,
3.  $f_1, f_2, g_1, g_2$  pedig leképezések.

Jelölje  $C$  a  $G$  gráf csúcspontjainak halmazát, és legyen  $C_1$  és  $C_2$  két idegen, nem üres részhalmaza  $G$ -nek. Ezek után  $f_1$  és  $f_2$  a  $C_1$  és  $C_2$  halmazok  $K$ -ra való leképezéseit jelentik.

Jelölje tovább  $E$  a  $G$  gráf éleinek halmazát, és legyen  $E_1$  és  $E_2$  két idegen,  $E_1$  nem üres, részhalmaza  $E$ -nek. Ezek után  $g_1$  az  $E_1$  halmaz leképezését  $S$ -re,  $g_2$  pedig  $E_2$  leképezését jelenti  $K$ -ba.

Mi — zászlós-ábráinknál — az alábbi elnevezéseket és szimbólum-rendszert alkalmazunk.

Az  $S$  halmaz  $s$  elmeit, illetve a  $K$  halmaz  $k$  elemeit továbbra is szavaknak, illetve kategóriáknak nevezzük.

A  $c_1 \in C_1$  csúcspontot, ha  $f_1(c_1) = k \in K$ , a  $k$  kategória kezdőpontnak, míg a  $c_2 \in C_2$  csúcspontot, ha  $f_2(c_2) = k \in K$  a  $k$  kategória végpontnak nevezzük.

Egy  $k$  kategória kezdő — és végpontját a  $G$  gráfban zászló-fejjel ábrázoljuk, mely balra, illetve jobbra mutat aszerint, hogy kezdő, illetve végpont jeléül szolgál. E zászlófej olyan derékszögű háromszög, mely egyik befogóján áll és a derékszög  $e$  befogón jobbra, illetve balra helyezkedik el (2. ábra):



2. ábra.

A zászlós-ábrák jobb áttekintése miatt a  $k$  kategória zászló-fejeit *nyéllel* láthatjuk el (3. ábra):



3. ábra 3

Az  $e_1 \in E_1$  élt  $(g_1(e_1) = s \in S)$   $s$  szó élnek nevezzük. A szó éleket a következő módon jelöljük (4. ábra):



4. ábra.

Végül az  $e_2 \in E_2$  élt  $(g_2(e_2) = k \in K)$   $k$  kategória élnek nevezzük. Egy  $k$  kategória élt olyan kettős zászló-fejjel ábrázolunk, mely balra is és jobbra is mutat. A kettős zászlós-fej olyan derékszögű háromszög, mely átfogóján áll (5. ábra):



5. ábra.

A *FORTRAN—PZ* nyelv ábrázolásakor a nyelv minden kategóriájához egy sorszámot is hozzárendelünk. Az ábrákon a kategóriákra a hozzájuk rendelt sorszámokkal hivatkozunk, azaz e sorszámot írjuk be a zászló-fejekbe. A zászlónyelek hosszának nem tulajdonítunk jelentőséget, csupán — esztétikai szempontból — az egy kategóriához tartozó kezdő — és vég zászló-fejekhez egyforma hosszúságú nyelet rajzolunk. A kettős zászló-fejekhez nyelet nem rajzolunk.

Az olyan zászlós-ábrát, mely  $k$  éleket nem tartalmaz ( $E_2$  üres halmaz) *véges állapotú*-nak nevezzük (egy ilyen, egy  $A = \{S, K, G, f_1, f_2, g_1\}$  rendezett hatos ad meg). Véges állapotú zászlós-ábrához a zászlós-ábra *deriváltjain* (leszármazottjain) keresztül juthatunk. Ha egy zászlós-ábra valamely  $k$  élét  $k$  *részgráfjával* helyettesítünk, akkor a zászlós-ábra egy deriváltját nyerjük. Egy zászlós-ábra bármely két  $c_1 \in C_1$  és  $c_2 \in C_2$  pontját összekötő  $G'$  gráfot  $G$  *részgráfjának* nevezzük,  $k$  részgráfról pedig akkor beszélünk, ha  $c_1$  és  $c_2$  pontokra teljesül, hogy  $f_1(c_1) = f_2(c_2) = k$ .

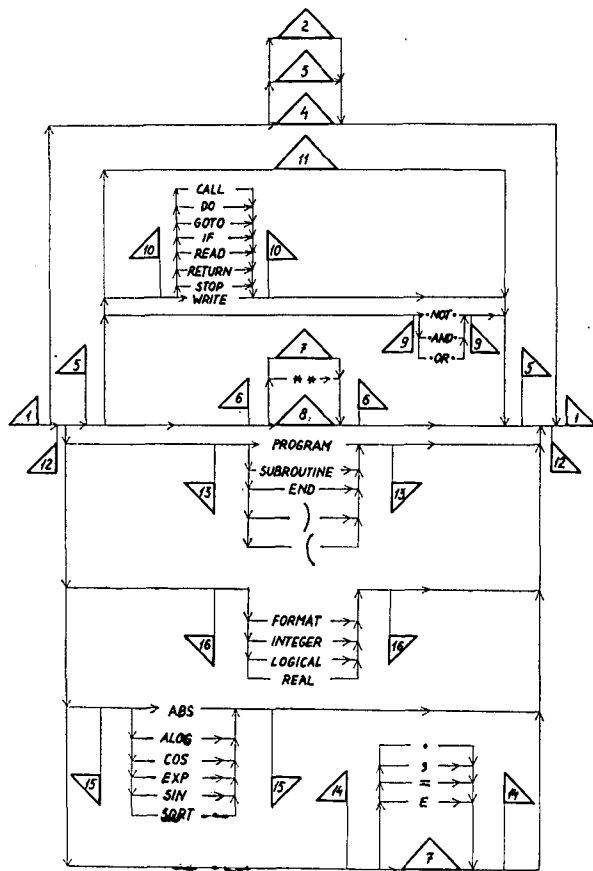
Ezek után

Definíció: egy  $L = \{S, K, f\}$  nyelvet az  $A = \{S, K, G, f_1, f_2, g_1, g_2\}$  zászlós-ábra által *reprezentált nyelvnek* nevezzük, ha  $f$  minden  $k$  kategóriához mindazon  $f(k)$  szó-sorozatok halmazát rendeli, melyek a zászlós-ábra valamely deriváltjában, valamely  $k$  él mentén leolvashatók.

A fent leírt fogalmak és jelölésrendszer alapján reprezentáljuk a *FORTRAN—PZ* nyelvet zászlós-ábrákkal.

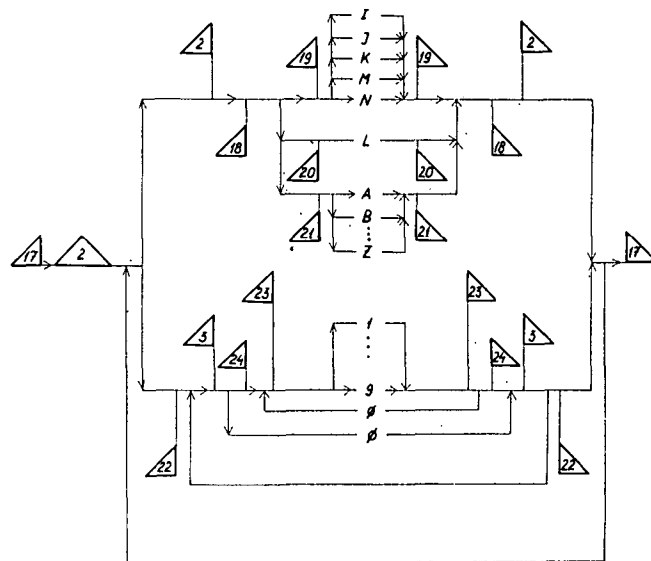
### A *FORTRAN—PZ* nyelv definíciója

A 6—20 ábra-sor adja meg a *FORTRAN—PZ* nyelvet. Az ábrák után azok a kategória-fogalmak találhatók, melyek kifejtései az illető ábrákon vannak.



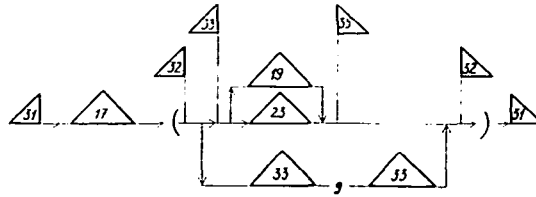
6. ábra.

- |                             |                           |
|-----------------------------|---------------------------|
| 1. Alapjel                  | 12. Elhatároló jel        |
| 5. Műveleti jel             | 13. Zárójel               |
| 6. Aritmetikai műveleti jel | 14. Elválasztó jel        |
| 9. Logikai műveleti jel     | 15. Standard-függvény jel |
| 10. Vezérlőjel              | 16. Deklaráló jel         |

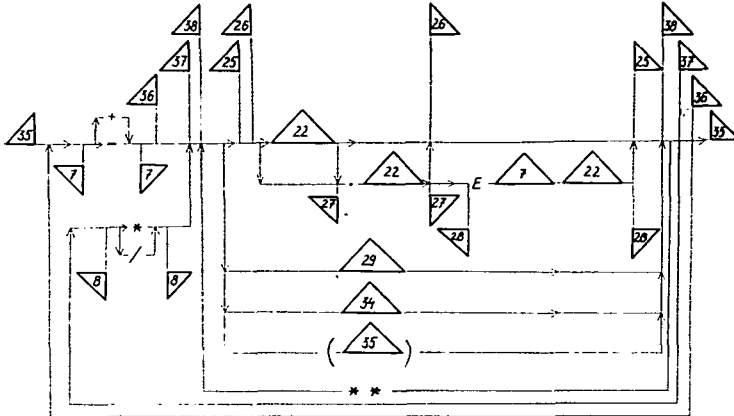


7. ábra.

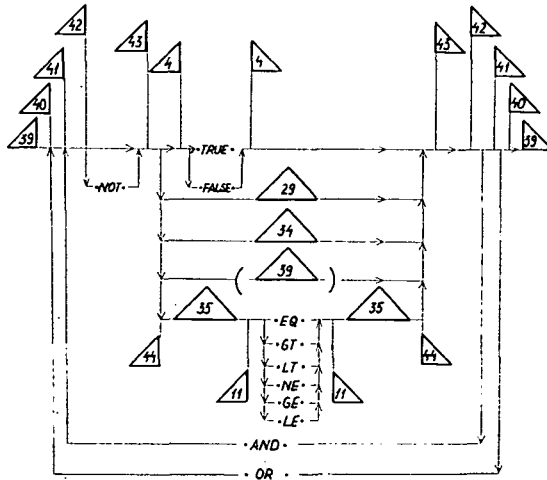
- |                            |                      |
|----------------------------|----------------------|
| 2. Betű                    | 20. Logikai betű     |
| 3. Számjegy                | 21. Többi betű       |
| 17. Azonosító              | 22. Természetes szám |
| 18. Automatikus deklaráció | 23. Értékes számjegy |
| 19. Egész betű             | 24. Címke.           |



8. ábra. 31. Indexes változó. 32. Indexlista. 33. Indexkifejezés

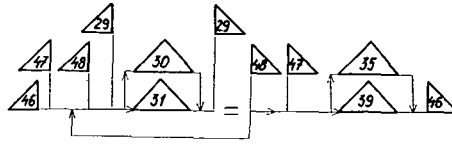


9. ábra. 7. Additív műveleti jel  
8. Multiplikatív műveleti jel  
25. Előjel nélküli szám  
26. Tizedesszám  
27. Valódi tizedestört  
28. Kitevőrész  
35. Aritmetikai kifejezés  
36. Tag  
37. Tényező  
38. Elsődleges kifejezés

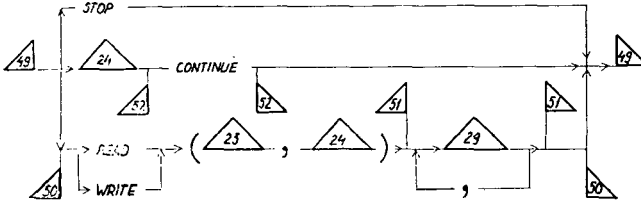


10. ábra. 5. Logikai érték  
11. Reláció jel  
39. Logikai kifejezés  
40. Diszjunkciós kifejezés  
41. Logikai tag  
42. Logikai tényező  
43. Elsődleges logikai kifejezés  
44. Reláció

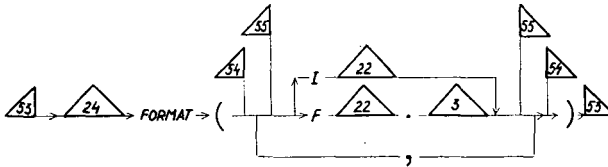




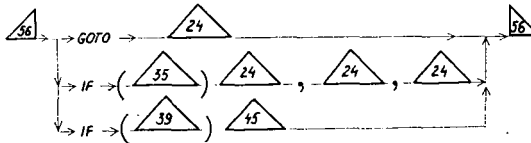
11. ábra. 29. Változó  
46. Értékdő utasítás  
47. Bal oldalak listája  
48. Bal oldal



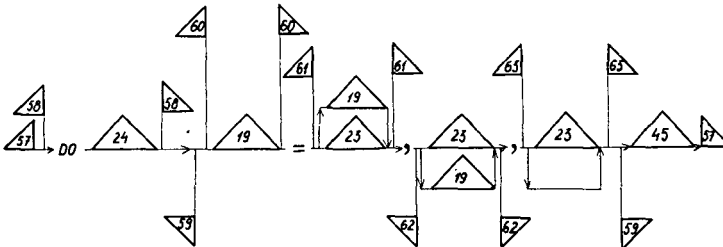
12. ábra. 49. Szervező utasítás  
50. I/o utasítás  
51. I/o lista  
52. Üres utasítás



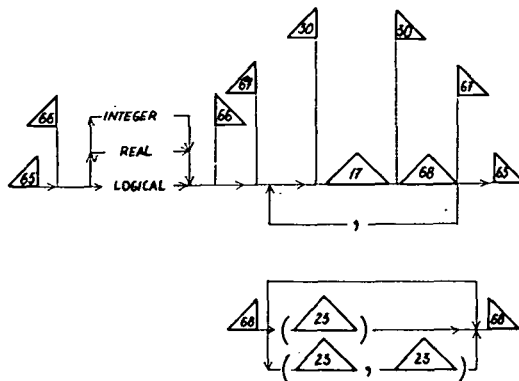
13. ábra: 53. FORMAT utasítás  
54. FORMAT lista  
55. FORMAT listaelem



14. ábra. 56. Vezérlő-utasítás

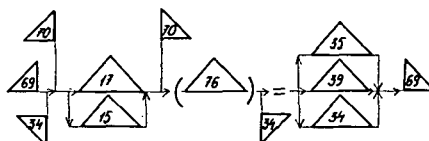


15. ábra. 57. Ciklusutasítás  
58. Ciklus kezdet  
59. Ciklus specifikáció  
60. Ciklus paraméter  
61. Kezdőérték  
62. Végérték  
63. Növekmény



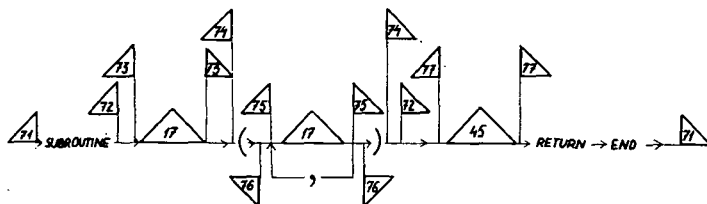
16. ábra. 30. Skaláris változó  
65. Típusdeklaráció

66. Típus  
67. Típuslista  
68. Dimenzió



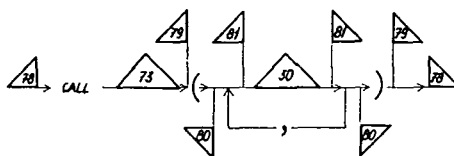
17. ábra. 34. Függvénykifejezés

69. Utasítás függvény  
70. Függvénynév.



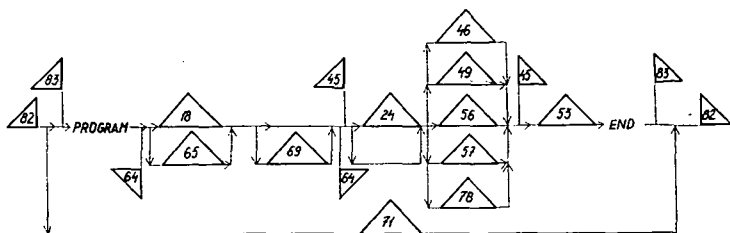
18. ábra. 71. Eljárás szegmens  
72. Eljárás fej  
73. Eljárásnév

74. Formális paraméter-rész  
75. Formális paraméter  
76. Formális paraméterlista  
77. Eljárástörzs



19. ábra. 78. Eljárás-utasítás  
79. Aktuális paraméter-rész

80. Aktuális paraméterlista  
81. Aktuális paraméter



20. ábra. 45. Utasítás  
64. Deklaráció

82. Szegmens  
83. Fő-szegmens

## IRODALOM

- [1] Az elektronikus, digitális számítógépek eddigi fejlődése és a várható fejlődése fő irányai (tanulmány), Szeged, 1972.
- [2] KALMÁR L.: Az intuitive representation of context-free languages, COLING, The proceedings of the International Conference on Computational Linguistics, Sanga-Söby, 1969.
- [3] PUSKÁS A.: Számítástechnikai képzés a tanárképző főiskolában (előadás), Szeged, 1971.
- [4] PUSKÁS A.: A FORTRAN programozási nyelv egy változata (előadás), Visegrád, 1974.
- [5] SZENTGYÖRGYI Zs.: Új Babel épül?, I—II. (A programozási nyelvek fejlődése), Természet Világa, 4, 6, 173—177, 263—266, 1974.

## ЯЗЫК ПРОГРАММИРОВАНИЯ FORTRAN-PZ

A. Пушкаш

В работе определяется один из видов языка FORTRAN, язык FORTRAN-PZ.

Автор при разработке варианта выбрал первичным аспектом усвояемость и обучаемость. Далее было его целью определить такой язык программирования, который способен знакомить участников основного курса расчётной кш техники с построением, структурой языка программирования.

Синтаксис языка автор определяет рисунками-флажками. Изображение синтаксиса языка рисунками с флажками можно выгодно употреблять как вспомогательное учебное пособие в обучении языка программирования.

## DIE FORTRAN-PZ-PROGRAMMIERUNGSSPRACHE

A. Puskás

Die Arbeit definiert eine Variante der FORTRAN-Sprache, die FORTRAN-PZ-Sprache.

Verfasser wählte bei der Ausarbeitung der Variante als leitenden Gesichtspunkt die Erlernbarkeit bzw. die Lehrbarkeit. Weiters verfolgte er das Ziel, eine Programmierungssprache zu definieren, die ausreicht, um den an der rechentechnischen Grundausbildung Teilnehmenden den Aufbau, die Struktur einer Programmierungssprache verständlich bzw. bekannt zu machen.

Die Syntax der Sprache erläuterte Verfasser mit Fahnenabbildungen. Die Lösung der Satzlehre einer Sprache mit Fahnenabbildungen ist nämlich als didaktisches Hilfsmittel im Unterricht der Programmierungssprache vorteilhaft brauchbar.